UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/069,670	02/22/2002	Xavier Leroy	102114.00034	3622

54975 7590 06/05/2009

HOLLAND & KNIGHT LLP
10 ST. JAMES AVENUE
BOSTON, MA 02116-3889

EXAMINER	
SWEARINGEN, JEFFREY R	

ART UNIT	PAPER NUMBER
2445	

MAIL DATE	DELIVERY MODE
06/05/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/069,670	LEROY, XAVIER	
	Examiner	Art Unit	
	Jeffrey R. Swearingen	2445	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 March 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *4-6,8-14,20,22 and 24-27* is/are pending in the application.

4a) Of the above claim(s) *15-18* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *4-6,8-14,20,22 and 24-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Election/Restrictions

1. Applicant's election with traverse of Group I in the reply filed on 3/6/2009 is acknowledged. The traversal is on the ground(s) that the restriction was made based solely on Applicant's statement that Claims 15-16 were not substantially the same as claim 4. This is not found persuasive because Applicant's statements revealed further differences in the claims.

"[I]ndependent claims 4, 15, and 16 include different limitations and, as such, it is Applicant's understanding that they are not "substantially the same". Remarks, 3/6/2009, page 21. "Applicant respectfully submits that method claims 15-17 may be used to guarantee that software modified by these methods will pass the validation steps of the claims of Group I when downloaded as it complies to test conditions such as 'no stack overflow', 'execution stack empty', etc." Remarks, 3/6/2009, page 21.

Applicant has provided further statements conceding that Method claims 15-17 (and dependent claim 18) of Group II are not necessary to the validation methods of Group I.

Applicant did not address the separate grouping of the claims presented in the restriction requirement of 1/6/2009. Applicant did not address the separate classification of Group I in 709/220 and Group II in 712/227 as presented in the restriction requirement of 1/6/2009. Applicant did not present evidence outside of the originally filed specification to rebut the restriction requirement or separate classification of Groups I and II.

The requirement is still deemed proper and is therefore made FINAL.

2. Claims 15-18 are withdrawn from further consideration pursuant to 37 CFR 1.142(b), as being drawn to a nonelected invention, there being no allowable generic or linking claim. Applicant timely traversed the restriction (election) requirement in the reply filed on 3/6/2009.

Response to Arguments

3. Applicant's arguments with respect to claims 4-6, 8-14, 20, 22, and 24-27 have been considered but are moot in view of the new ground(s) of rejection.

Specification

4. The disclosure is objected to because of the following informalities: Claims 24-27 are directed toward a computer program product recorded on a medium. There is no definition in the specification for the medium upon which the computer program product is recorded.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claim 22 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claim 22 is directed to a system for transforming an objected code of a program fragment. The system includes a program module for performing the transformation steps of claim 22. Claim 22 states the program module is "installed in a working memory of a development computer or

workstation." The term "working memory" is not readily known in the art. It is unclear if Applicant is intending to just encompass a program on physical memory, or if Applicant is attempting to broaden the scope of "working memory" to encompass software *per se*. Software *per se* is not patentable subject matter.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 4, 8, 12, 13, 20, 22 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

9. Claim 4 has an execution stack in the preamble and a type stack in the first initializing clause. The carrying out clause verifies "that said stack is empty". It is indefinite whether "said stack is empty" refers to the execution stack or the type stack.

10. It is unclear what Applicant is attempting to claim in claim 6.

11. Claim 8 refers to "said stack of variable types". It is indefinite whether "said stack of variable types" refers to the execution stack or the type stack.

12. Claim 12 refers to "said type execution stack." It is indefinite whether "said type execution stack" refers to the execution stack or the type stack. Claim 12 further refers to "said stack", which may refer to either the execution stack or the type stack.

13. Claim 13 refers to "the stack". It is unclear if this is a reference to "said type stack" in the same clause.

14. Claim 20 refers to "said stack is empty". It is indefinite whether "said stack is empty" refers to the type stack.

15. Claim 22 refers to "said data type of said stack". It is indefinite whether "said stack" refers to the execution stack or the type of stack variables.

Claim Rejections - 35 USC § 103

16. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. Claims 4, 8-14, 20, 22 and 24-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yellin et al. (US 5,740,441) in view of Nagel (US 6,071,317).

18. In regard to claim 4, Yellin disclosed *a method of verifying a program fragment downloaded onto a reprogrammable embedded system, equipped with a rewritable memory, a microprocessor and a virtual machine equipped with an execution stack and with operand registers, said program fragment consisting of an object code and including at least one subprogram consisting of a series of instructions manipulating said operand registers, said microprocessor and virtual machine configured to interpret said object code, said embedded system being interconnected to a reader, wherein subsequent to a detection of a downloading command and a storage of said object code in said rewritable memory, said memory, for each subprogram, comprises:*

initializing a type stack and a table of register types through data representing a state of said virtual machine on initialization of an execution of said temporarily stored object code; Yellin, column 8, lines 41-44

carrying out a verification process of said temporarily stored object code instruction by instruction, by discerning an existence, for each current instruction, of a target, a branching-instruction target, a target of an exception-handler call or a target of a subroutine call, and, said current instruction being a target of a branching instruction, said verification process including verifying that said stack is empty and rejecting said program fragment otherwise; Yellin, column 9, line 61 - column 10, line 17

verifying and updating an effect of said current instruction on said data types of said type stack and of said table of register types; Yellin, column 10, lines 18-34

said verification process being successful when said table of register types is not modified in the course of a verification of all said instructions, and said verification process being carried out instruction by instruction until said table of register types is stable, with no modification being present, Yellin, column 8, lines 52-56 *said verification process being interrupted and said program fragment being rejected, otherwise.* Yellin, column 8, lines 57-61

Yellin failed to disclose a verification process on an embedded system using a virtual machine. However, Nagel disclosed in column 14, lines 56-66 the implementation of a similar system to Yellin in an embedded system. Nagel disclosed

implementation using a virtual machine implemented on specific hardware. Nagel, column 11, lines 61-67. Nagel also detected object code alterations in column 23, lines 41-57.

It would have been obvious to one of ordinary skill in the art at the time of invention to implement the Yellin object code verification system on an embedded system such as Nagel in order to allow for debugging of code in embedded systems where programming is not as easily altered by one of ordinary skill in the art during normal usage.

19. In regard to claim 8, Yellin disclosed *said current instruction being a target of a subroutine call, said verification process comprising:*

verifying that a previous instruction to said current instruction is an unconditional branching, a subroutine return or a withdrawal of an exception; and reupdating said stack of variable types by an entity of the return address type, formed by said return address of the subroutine, in case of a positive verification process; and, Yellin, column 9, lines 30-52

rejecting said program fragment in case said verification process is failing, otherwise. Yellin, column 9, lines 53-60

20. In regard to claim 9, Yellin disclosed *said current instruction being a target of an exception handler, said verification process comprising:*

verifying that a previous instruction to said current instruction is an unconditional branching, a subroutine return or a withdrawal of an exception; and Yellin, column 11, lines 1-36

reupdating said type stack, by entering an exception type, in case of a positive verification process; and Yellin, column 11, lines 1-36

rejecting said program fragment in case of said verification process is failing. Yellin, column 11, lines 1-36

21. In regard to claim 10, Yellin disclosed *said verification process fails and said program fragment is rejected if said current instruction is said target of multiple incompatible branchings.* Yellin, column 7, lines 25-39

22. In regard to claim 11, Yellin disclosed *said verification process comprises continuing by passing to an update of said type stack if said current instruction is not said target of any branching.* Yellin, column 7, lines 56-61

23. In regard to claim 12, Yellin disclosed *said verifying and updating includes, at least:*

verifying that said type execution stack includes at least as many entries as a current instruction includes operands; Yellin, column 9, lines 30-36

unstacking and verifying that types of entries at a top of said stack are subtypes of types of operands of said operands of said current instruction; Yellin, column 9, lines 40-52

verifying an existence of a sufficient memory space on said type stack to proceed to stack said results of said current instruction; Yellin, column 10, lines 18-42

stacking data types which are assigned to the results on said stack. Yellin, column 10, lines 15-17

24. In regard to claim 13, Yellin disclosed *said current instruction being an instruction to read a register of a given address, said verification process comprises:*

verifying said data type of the result of a corresponding reading, by reading an entry at said given address in said table of register types; Yellin, column 10, lines 18-34

determining said effect of said current instruction on said type stack by unstacking said entries of the stack corresponding to the operands of said current instruction and by stacking said data type of said result. Yellin, column 10, lines 18-34

25. In regard to claim 14, Yellin disclosed *said current instruction being an instruction to write to a register of a given address, said verification process comprises:*

determining an effect of said current instruction on said type stack and said given type of said operand which is written in this register at said given address; Yellin, column 10, lines 49-60

replacing said type entry of said table of register types at said given address by said type immediately above said previously stored type and above said given type of said operand which is written in said register at said given address. Yellin, column 10, lines 49-60

26. Claim 20 is substantially the same as claim 4.

27. In regard to claim 22, Yellin disclosed *a system for transforming an object code of a program fragment including a series of instructions, in which operands of each instruction belong to data types manipulated by said instruction, an execution stack*

does not exhibit any overflow phenomenon and for each branching instruction, a type of stack variables at a corresponding branching is identical to the targets of this branching, and said operand, of given type, written to a register by an instruction of said object code is reread from said same register by another instruction of said object code with the same given data type, into a standardized object code for said same program fragment, wherein said transforming system includes, at least, installed in a working memory of a development computer or workstation, a program module for transforming said object code into a standardized object code in accordance with a process of transforming including for all said instructions of said object code comprising:

annotating each current instruction with said data type of said stack before and after execution of said current instruction, with an annotation data being calculated by means of analysis of the data stream relating to said current instruction; Yellin, column 10, line 53 – line 66

detecting, within said instructions and within each current instruction, an existence of branchings, or respectively of branching-targets, for which said execution stack is not empty, said detecting operation being carried out on a basis of said annotation data of said type of stack variables allocated to each current instruction; Yellin, column 10, lines 43-48 *and, in case of detection of a non-empty execution stack,*

inserting instructions to transfer stack variables on either side of said branchings or of said branching targets respectively, in order to empty contents of said execution stack into temporary registers before said branching and to

reestablish said execution stack form said temporary registers after said branching; and Yellin, column 10, line 61 – column 11, line 30

not inserting any transfer instruction otherwise, said method allowing thus to obtain said standardized object code for said same program fragment, in which said operands of each instruction belong to said data types manipulated by said instruction, said execution stack does not exhibit any overflow phenomenon, said execution stack is empty at each branching instruction and at each branching-target instruction, in an absence of any modification to an execution of said program fragment. Yellin, column 10, line 61 – column 11, line 30

Yellin failed to disclose a verification process on an embedded system using a virtual machine. However, Nagel disclosed in column 14, lines 56-66 the implementation of a similar system to Yellin in an embedded system. Nagel disclosed implementation using a virtual machine implemented on specific hardware. Nagel, column 11, lines 61-67. Nagel also detected object code alterations in column 23, lines 41-57.

It would have been obvious to one of ordinary skill in the art at the time of invention to implement the Yellin object code verification system on an embedded system such as Nagel in order to allow for debugging of code in embedded systems where programming is not as easily altered by one of ordinary skill in the art during normal usage.

28. Claim 24 is substantially the same as claim 4.

29. Claim 25 is substantially the same as claim 22.

30. In regard to claim 26, Yellin disclosed *a computer program product which is recorded on a medium and can be used in a reprogrammable embedded system, equipped with a microprocessor and a rewritable memory, said reprogrammable embedded system allowing to download a program fragment consisting of an object code, a series of instructions, executable by said microprocessor of said reprogrammable embedded system by means of a virtual machine equipped with an execution stack and with local variables or registers manipulated via instructions and making it possible to interpret said object code, said computer program product comprising:*

program resources which can be read by said microprocessor of said embedded system via said virtual machine, to command execution of a procedure for managing a downloading of a downloaded program fragment; Yellin disclosed the verification system was applicable to downloaded program fragments in the prior art. Yellin, column 1, lines 33-54

program resources which can be read by said microprocessor of said embedded system via said virtual machine, to command execution of a procedure for verifying, by instruction, said object code which makes up said program fragment; Yellin, column 3, lines 46-59

program resources which can be read by said microprocessor of said embedded system via said virtual machine, to command execution of a downloaded program fragment subsequent to or in an absence of a conversion

of said object code of said program fragment into a standardized object code for this same program fragment. Yellin, column 3, lines 46-59

Yellin failed to disclose a verification process on an embedded system using a virtual machine. However, Nagel disclosed in column 14, lines 56-66 the implementation of a similar system to Yellin in an embedded system. Nagel disclosed implementation using a virtual machine implemented on specific hardware. Nagel, column 11, lines 61-67. Nagel also detected object code alterations in column 23, lines 41-57.

It would have been obvious to one of ordinary skill in the art at the time of invention to implement the Yellin object code verification system on an embedded system such as Nagel in order to allow for debugging of code in embedded systems where programming is not as easily altered by one of ordinary skill in the art during normal usage.

31. In regard to claim 27, Yellin disclosed *at least one program resources which can be read by said microprocessor of said embedded system via said virtual machine, to command inhibition of execution, by said embedded system, of said program fragment in the case of an unsuccessful verification procedure of said program fragment.* Yellin, column 4, lines 20-41

32. Claims 5-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yellin in view of Nagel as applied to claim 4 above, and further in view of Blume (US 6,223,337) and Haley et al. (US 5,694,539) and Chan et al. (US 5,276,881).

33. In regard to claims 5 and 6, Yellin disclosed *said variable types which are manipulated during said verification process include at least:*

class identifiers corresponding to object classes which are defined in said program fragment; Yellin, column 6, lines 59-64

numeric variable types including at least a type for an integer coded on a given number of bits, designated as short type, and a type for the return address of a jump instruction, designated as a return address type; Yellin, column 6, lines 15-22

object type relating to objects designated as object type; Yellin, column 6, lines 59-66

34. Yellin and Nagel failed to disclose the presence of a null type, a union type and an intersection type. Haley disclosed use of a NULL type. Haley, column 2, line 41, column 10, line 27. Blume disclosed an intersection type. Blume, column 7, Table 3. Chan disclosed a union type. Chan, column 32, line 21. It would have been obvious to one of ordinary skill in the art at the time of invention to verify any object type, including NULL, intersection, or union, with the Yellin/Nagel verification system.

Conclusion

35. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Jacobson	US 6,195,774
Wahbe et al.	US 6,151,618

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jeffrey R. Swearingen whose telephone number is (571)272-3921. The examiner can normally be reached on M-F 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Vivek Srivastava can be reached on 571-272-7304. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jeffrey R. Swearingen
Examiner
Art Unit 2445

/J. R. S./
Examiner, Art Unit 2445

/VIVEK SRIVASTAVA/

Supervisory Patent Examiner, Art Unit 2445